

「 게임 프로그래밍 」

HTML5 + CAVAS

# 코드 업그레이드

캔버스를 이용한 장애물 피하기 게임

「 목차 」

# 목차

1

코드 시범

2

추가된 기능 코드 설명

3

코드 정리

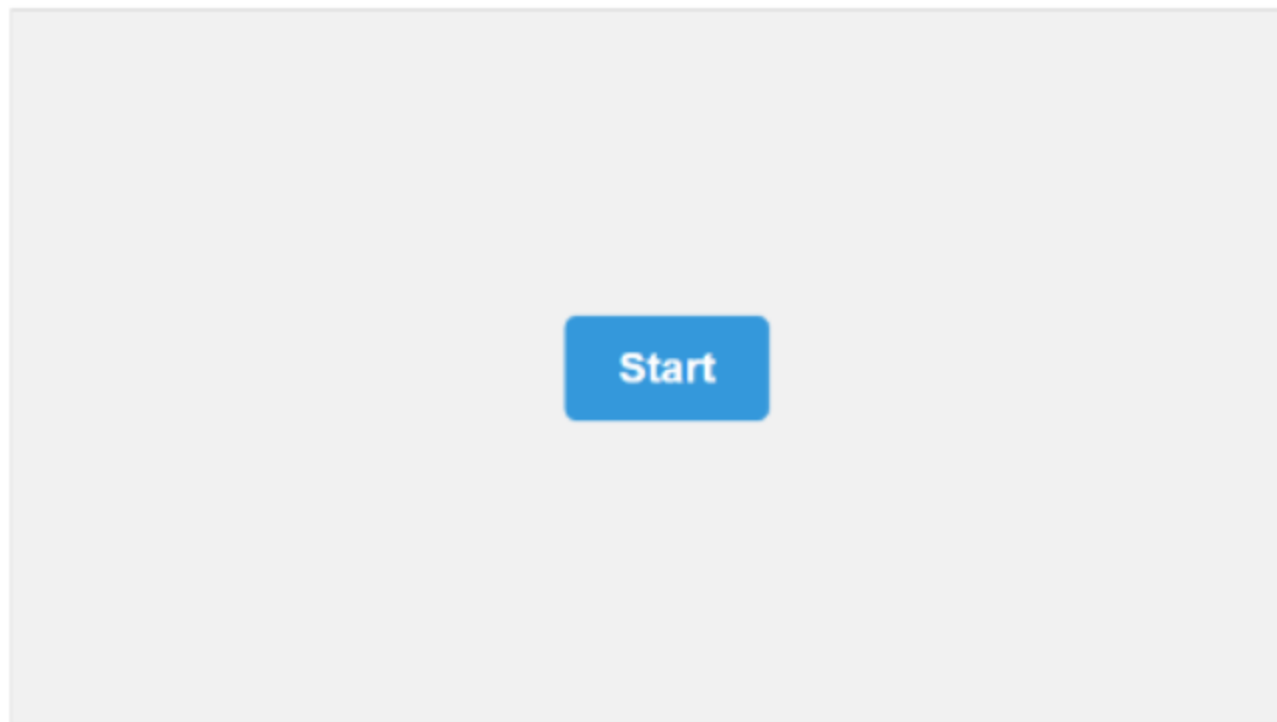


## 추가된 기능 코드 설명

## INTRO

## 화면 조정 및 인트로 생성

redyGame() 함수 도입



```
function readyGame() {  
    myBackground = new component(656, 270, Background[0], 0, 0, "background");  
    mySound = new sound("픽.mp3");  
    myMusic = new sound("BGM1.mp3");  
    myMusic.play();  
    myGameArea.start();  
}  
  
function startGame() {  
    myheart = new component(30, 26, "heart.png", 480, 130, "image");  
    document.getElementById("start-button").style.display = "none";  
    myGamePiece = new component(33, 16, "/image/turtle/1.png", 10, 120, "image");  
}
```

추가된 기능 코드 설명

# 방향키 사용

## 키보드 핸들러 도입

마우스 클릭은 게임에 부적합함. 이를 해결하기 위하여  
document.addEventListener() 사용



```
function moveup() { myGamePiece.speedY = -1; }  
function movedown() { myGamePiece.speedY = 1; }  
function moveleft() { myGamePiece.speedX = -1; }  
function moveright() { myGamePiece.speedX = 1; }
```

5

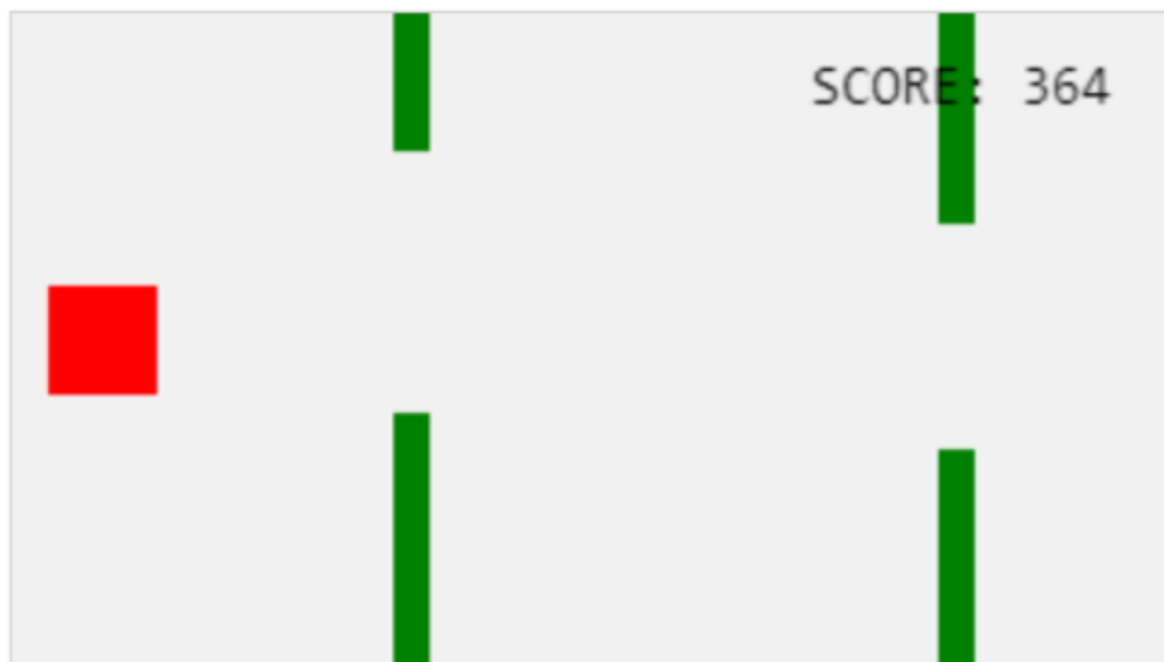
```
document.addEventListener("keydown", keyDownHandler, false);  
document.addEventListener("keyup", keyUpHandler, false);
```

```
function keyDownHandler(e) {  
    if (e.key == "ArrowUp") {  
        moveup();  
    } else if (e.key == "ArrowDown") {  
        movedown();  
    } else if (e.key == "ArrowLeft") {  
        moveleft();  
    } else if (e.key == "ArrowRight") {  
        moveright();  
    }  
}
```

```
function keyUpHandler(e) {  
    if (e.key == "ArrowUp" || e.key == "ArrowDown") {  
        clearmove();  
    }  
    if (e.key == "ArrowLeft" || e.key == "ArrowRight") {  
        clearmove();  
    }  
}
```

추가된 기능 코드 설명

# 에셋의 활용



## 추가된 기능 코드 설명

```
myGamePiece = new component(33, 16, "/image/turtle/1.png", 10, 120, "image");  
myBackground = new component(656, 270, Background[0], 0, 0, "background");
```

```
this.update = function () {  
    ctx = myGameArea.context;  
    if (type == "image" || type == "background") {  
        ctx.drawImage(this.image,  
            this.x,  
            this.y,  
            this.width, this.height);  
        if (type == "background") {  
            ctx.drawImage(this.image,  
                this.x + this.width,  
                this.y,  
                this.width, this.height);  
        }  
    } else {  
        ctx.fillStyle = color;  
        ctx.fillRect(this.x, this.y, this.width, this.height);  
    }  
}
```

## Component 생성자 활용

new component(가로크기, 세로크기, 이미지 주소,  
X좌표, Y좌표, 이미지 타입)








## 추가된 기능 코드 설명

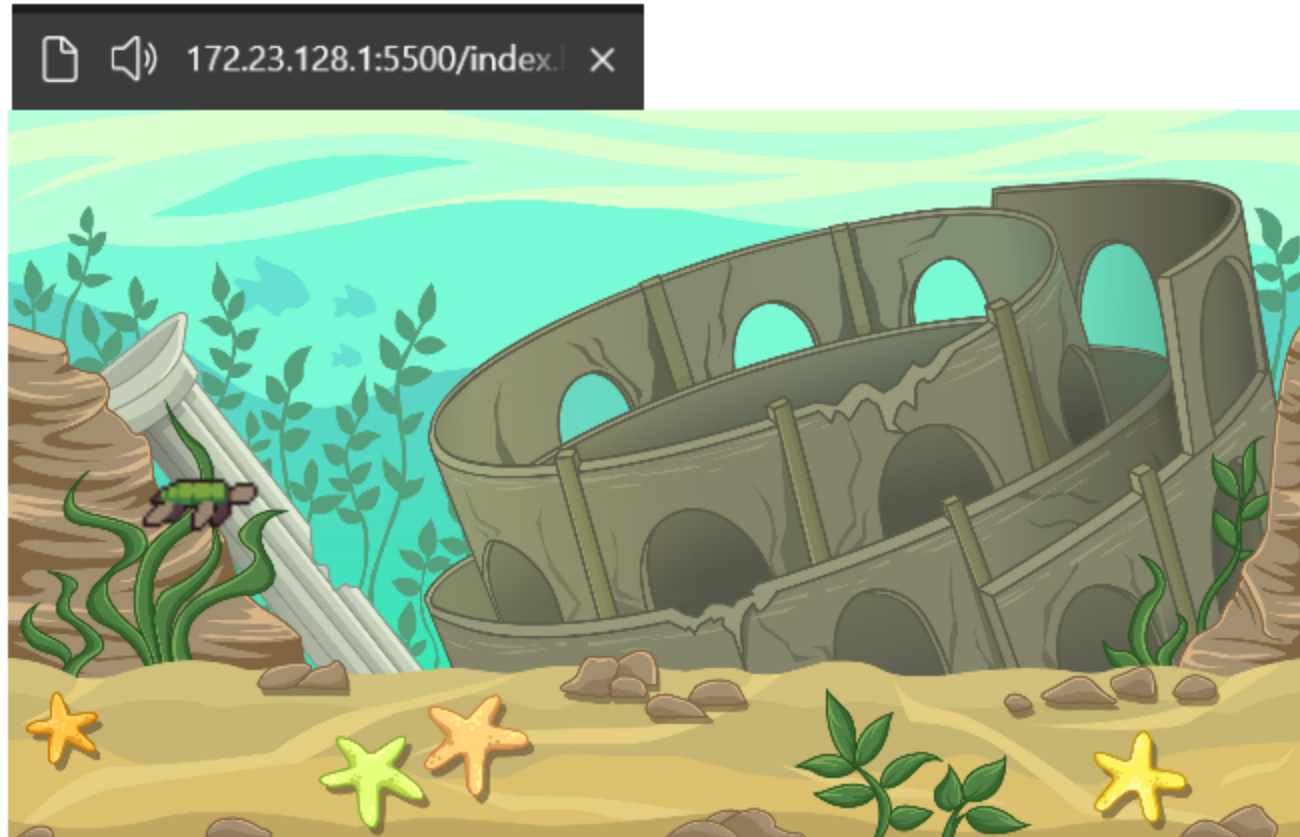
```
function readyGame() {  
    myBackground = new component(656, 270, Background[0], 0, 0, "background");  
    mySound = new sound("픽.mp3");  
    myMusic = new sound("BGM1.mp3");  
    myMusic.play();  
    myGameArea.start();  
}
```

```
function sound(src) {  
    this.sound = document.createElement("audio");  
    this.sound.src = src;  
    this.sound.setAttribute("preload", "auto");  
    this.sound.setAttribute("controls", "none");  
    this.sound.style.display = "none";  
    document.body.appendChild(this.sound);  
    this.play = function(){  
        this.sound.play();  
    };  
    this.stop = function(){  
        this.sound.pause();  
    };  
}
```

 BGM1.mp3  
 BGM2.mp3  
 BGM3.mp3

## sound(src) 사용

```
this.sound = document.createElement("audio");
```

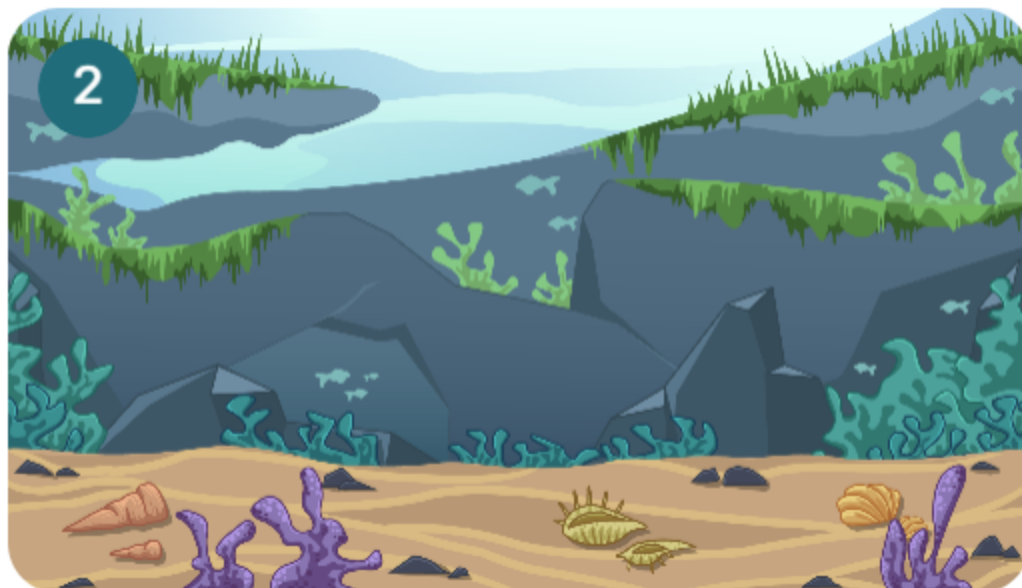


추가된 기능 코드 설명

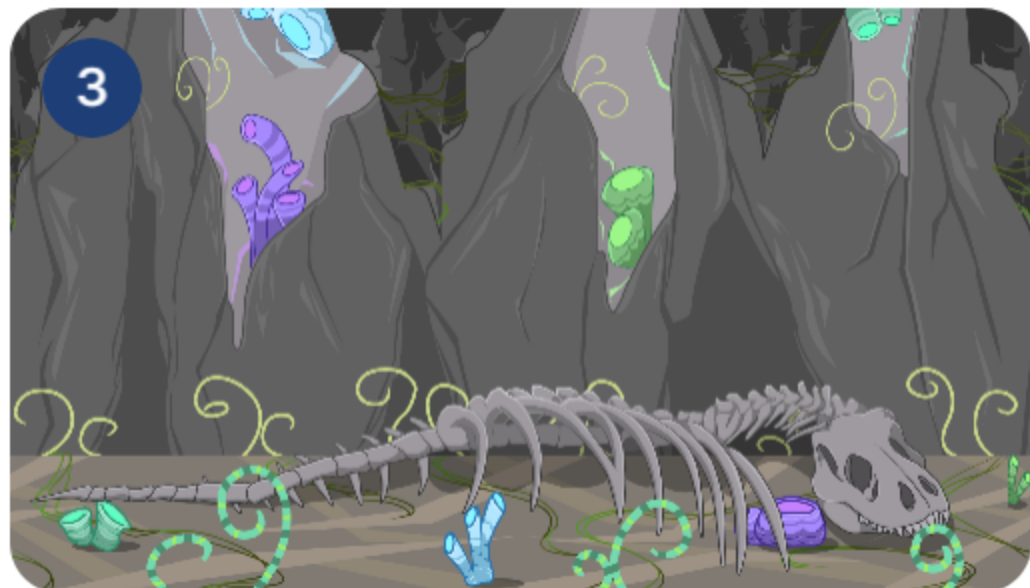
# 3가지 Stage

**Stage 1**

```
EVERYINTERVAL(150)  
MYOBSTACLES[I].SPEEDX = -1;
```

**Stage 2**

```
EVERYINTERVAL(100)  
MYOBSTACLES[I].SPEEDX = -3;
```

**Stage 3**

```
EVERYINTERVAL(90)  
MYOBSTACLES[I].SPEEDX = -5;
```



추가된 기능 코드 설명

# 랜덤하게 호출되는 방해물

1



갯지렁이

2



청새치

3



해파리



```
randomValue = Math.floor(Math.random() * 3);
```

추가된 기능 코드 설명

# 플레이어 목숨 추가



추가된 기능 코드 설명

# 플레이어 목숨 추가

1



생명력  
이미지 추가

2



데미지 감지 및  
함수 생성

3



일정기간 무적

4



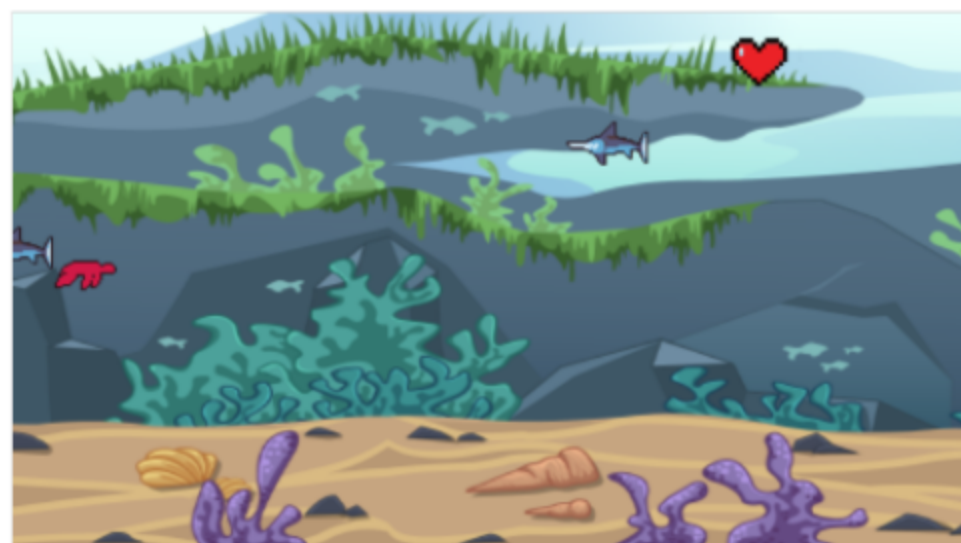
가시적 효과

## 추가된 기능 코드 설명

### 데미지 감지 함수

- 데미지 감지 후 heart의 값 줄이기
- 부딪히는 효과음 넣기

```
if(AUheart>0) {AUheart= AUheart-2;
}else if(AUheart == 0){
for (i = 0; i < myObstacles.length; i += 1) {
    if (myGamePiece.crashWith(myObstacles[i])) {
        AUheart = stage;
        mySound.play();
        heart = heart - 1;
    }
}}
```



### 이미지 변경 및 게임 중단



```
function damage() {
    if (heart > 0) {
        var heartImageSrc = "/image/heart" + heart + ".png";
        var heartComponent = new component(120, 50, heartImageSrc, 350, 0, "image");
        heartComponent.update();
    }
    if (heart == 0) {
        myGameArea.stop();
        document.getElementById("restart-button").style.display = "block";
        myMusic.stop();
        endGame();
        return;
    }
}
```



```
this.hurtupdate = function () {
    ctx = myGameArea.context;
    if (type == "image" || type == "background") {
        var hurtImage = new Image();
        hurtImage.src = "/image/turtle/7.png";
        ctx.drawImage(hurtImage,
            this.x,
            this.y,
            this.width, this.height);
    } else {
        ctx.fillStyle = color;
        ctx.fillRect(this.x, this.y, this.width, this.height);
    }
}
```

추가된 기능 코드 설명

# 회복 아이템 추가



```
if (myGamePiece.crashWith(myheart)) {  
    if(heart!=3){ heart+=1;}  
    myheart.speedX = -400;  
    myheart.newPos();  
    myheart.update();  
}
```

```
function damage() {  
    if (heart > 0) {  
        var heartImageSrc = "/image/heart" + heart + ".png";  
        var heartComponent = new component(120, 50, heartImageSrc, 350, 0, "image");  
        heartComponent.update();  
    }  
}
```



## 추가된 기능 코드 설명

## OUTTRO

## 게임 완료 버튼 및 재시작 버튼

restartGame() 함수 도입



```
function restartGame() {  
    myMusic.stop();  
    document.getElementById("restart-button").style.display = "none";  
    myGameArea.clear();  
    myObstacles = [];  
    myGameArea.frameNo = 0;  
    ester = 0;  
    document.getElementById("p").style.display = "none";  
    document.getElementById("com-button").style.display = "none";  
    heart = 3;  
    AUheart = 0;  
    stage = 150;  
    myBackground = new component(656, 270, Background[0], 0, 0, "background");  
    myGamePiece = new component(33, 16, "/image/turtle/1.png", 10, 120, "image");  
    myheart = new component(30, 26, "heart.png", 480, 130, "image");  
    mySound = new sound("프릭.mp3");  
    myMusic = new sound("BGM1.mp3");  
    myMusic.play();  
    myGameArea.start();  
}
```

추가된 기능 코드 설명

# 이스터 에그?



## 변수 ester

일정 시간 이상 화면 밖을 나가 있으면 발동되는 이벤트

```
if (ester>200) {document.getElementById("p").style.display = "block";}
myGamePiece.newPos();
if(AUheart == 0){myGamePiece.update();}
else{myGamePiece.hurtupdate();}
damage();
```

# 추가 기능 모음

U

「기존 기능 업데이트」

RESTART, NEXT 버튼 수정

플레이어 이미지 변경

배경 이미지 변경

장애물 종류 추가 및 방식 변경

조작키 변경

캔버스 화면 배율 조정

N

「새로운 기능 추가」

INTRO, OUTRO 추가

스테이지 추가 및 난이도 조절

목숨 기능 추가

배경을 및 효과음 추가

이스터 에그 추가

아이템 기능 추가

「 게임 프로그래밍 발표 」

# 감사합니다

<https://kenney.nl/assets>

<https://www.sellbuymusic.com/albumDetail/113>

김정원

2021863022

2023-11-15

